A Neural Network Structure for Detecting Straight Line Segments

Nabeel Murshed
Center for Information Science and Processing
Tuiuti University of Paraná
Curitiba, Brazil
nmurshed@cognus.eti.br, nmurshed@utp.br

Abstract

A new method for detecting one-pixel wide vertical, horizontal and diagonal line segments in binary images, is presented. It is based on using four slabs of neural networks, each of which is composed of a set layers. Each layer consists of a number of neurons that is determined by the slab type. The whole image is used as input to each slab, and the information processing in each slab occurs in parallel, decreasing, therefore, computation time and allowing hardware implementation. The method was tested with various types of binary images and the obtained results were satisfactory. In addition, the method was robust against random noise, such as straight lines impeded in a cloud of points.

1. Introduction

In image processing and pattern recognition the detection of lines is an important process in the classification of visual forms. Consider for example the hypothetical image shown in figure 1, which is composed of three rectangles and one triangle. In order to recognize these shapes one need to use, for example, line-tracing algorithms, sliding windows, or morphological filters [1] to detect the constituents (line segments) of each shape and, thereafter, use structural techniques to build the final decision. When using these techniques, the input image is scanned pixel by pixel and a set of sequential operations are performed. These sequential operations could become computationally expensive. The work presented here is concerned with developing detection method where information processing could be performed in parallel, and hardware implementation would be possible.

During the last three decades, numerous problems in science, technology, medicine, and finance have been solved using neural networks technology. In image processing neural networks have been applied for edge detection [2,3] and edge refinement [4,5]. In principle, the operation philosophy in each method was based on mapping gray-level image blocks, taken from the input image, into their most likely corresponding or refinement patterns using neural networks. The main advantage of using neural networks in those methods, is that processing can be done in parallel and, thus, computation time can be reduced. In this work we extend neural network technology to solving the problem of detecting line segments in binary images. Specifically, we employ the concepts of lateral excitation/inhibition [6], the 2/3 rule [7,8].

The proposed method is based on using four slabs of neural networks for detecting one-pixel-wide horizontal, vertical and diagonal lines line segments (—,|, /,\ ), in binary images. It is part of a project aimed at developing Neural Systems for recognizing visual text. For the sake of discussion, we will symbolize these lines by *h*, *v*, *d1* and *d2*, respectively. Accordingly, the slabs will be symbolized by *h_slab*, *v_slab*, *d1_slab* and *d2_slab*. The philosophy behind the proposed method is that horizontal, vertical, and diagonal lines in the input image are considered layers of neurons. Adjacency of active neurons in any layer is enforced, and scattered active neurons are eliminated. Information processing in each layer occurs in parallel.

Section 2 describes the network structure. Experimental results are given in section 3. Discussion and conclusion are given in section 4.

2. Network Structure[1]

Each slab is composed of a set of layers, each of which contains specific number of neurons. The layers in any given slab are oriented to detect one type of line segment only. For example in the horizontal and vertical slabs, the layers are oriented horizontally and vertically to detect horizontal and vertical lines, respectively. Similarly, in the diagonal slabs the layers are oriented diagonally to detect diagonal lines. Figure 2 gives an example of the structure of each slab. With such a structure, a given input image

[1] The network structure presented in this paper is registered under a Brazilian Patent Pending.

0-7803-5529-6/99/$10.00 ©1999 IEEE

will be decomposed into horizontal, vertical and diagonal lines, if these line are present in the image.

The size of each slab equals that of the input image. Consequently, the number of layers in the vertical and horizontal slabs equals, respectively, the number of columns and lines in the input image; and the number of neurons in a vertical/horizontal layer equals the number of pixels in a column/line. For the diagonal slabs, the number of layers equals the possible number of diagonal lines in the image; and the number of neurons in a given diagonal layer equals the number of pixels connected by the corresponding diagonal line in the image.

2.1 Neuron Structure

Each neuron in any given layer, except those at the extreme, is connected to its two neighboring neurons through bi-directional lateral connections of constant weights. Extreme neurons are connected to one neighboring neuron only. In addition, each neuron receives an input signal from the image pixel. The signal is excitatory, 1-valued, if it comes from a black pixel, and it is inhibitory, 0-valued, if it comes from a white pixel. Figure 3 illustrates the connections between neurons in a given layer. Note that there are no connections between neurons of different layers.

The information processing in each neuron occurs as follows: The two inputs from the neighboring neurons are summed up and an analog signal is generated. If the analog signal exceeds a threshold value, then a discrete internal signal is produced. Thereafter, the discrete internal signal is multiplied by the discrete external signal arriving from the image pixel. If both signal are excitatory, then the neuron fires. If either one or both are inhibitory, the neuron shuts off or remains inactive. A mathematical model of the neuron is illustrated in figure 4. The equations describing the neuron's internal information processing are given bellow:

$$y_a = \sum_{i=1}^{2} w_i x_i + b \quad (1)$$

$$y_d = \begin{cases} 1 & y_a \geq 1 \\ 0 & y_a = 0 \end{cases} \quad (2)$$

$$z = I_0 \times y_d \quad (3)$$

where x_i is the ith input signal from a neighboring neuron, and w_i is the weight connection and it equals 1; b is the bias input; y_a and y_d are, respectively, the internal analog and discrete signals; and z is the neuron's discrete output.

2.2 Network Operation

When an input image is presented, it is projected to each slab such that each neuron, in a slab, is connected to one pixel in the image. Consequently, each slab becomes a replica of the input image. Thereafter, each slab self-organizes its structure so as to detect one type of line segment only. The network operation is summarized bellow:

1. ***start;***
2. *present an input image of size nxm;*
3. *construct four matrices, each of size nxm;*
4. *transform each matrix into one type of slab;*
5. *connect a pixel p(l, k), in the image to a corresponding neuron, N(l, k), in each slab. Initially, all neurons are inactive.*

Stage 1:

6. *set the bias input of each neuron to 1;*
7. ***for each slab;*** *(em parallel)*
8. ***for each layer;*** *(em parallel)*
9. ***for each neuron;*** *(sequentially)*
10. *calculate the output;*
11. ***end for;***
12. ***end for;***
13. ***end for;***
14. *Note that at this stage each slab becomes a replica of the input image;*
15. ***end for;***
16. *set the bias input of each neuron to 0;*

Stage 2:

17. ***for each slab;*** *(em parallel)*
18. ***for each layer;*** *(em parallel)*
19. ***for each neuron;*** *(sequentially*
20. *calculate the output;*
21. ***end for;***
22. *repeat steps 19 to 21 until the output of each neuron does not change;*
23. ***end for;***
24. ***end for;***
25. ***end.***

To illustrate the network operation, consider the simple image shown in figure 5a, which consists of horizontal and vertical lines. After the execution of steps 1 through 16, each neuron in any slab will be activated/forced to remain inactive by the input signal coming from the image pixel. Consequently, each slab becomes a replica of the input image, as shown in figure 5b. Note that In digital image processing, a one-pixel wide line segment is a sequence of adjacent black pixels, in a given orientation. Similarly, in neural-based image processing one may define a line segment as a sequence of adjacent active neurons in any given layer.

During the next stage of the network operation (steps 17 through 24), each layer in a given type of slab self-organizes its structure such that it detects the same type of line segment. If a given type of line segment is not present in the image, then all the neurons of the corresponding slab becomes inactive. Figure 5c shows the final state of each slab.

2.3 Preservation of Line Connectivity and Noise Elimination

There are two principal rules governing the operation of the proposed network, at the layer level. The first rule is called *Preservation of Connectivity(POC)*, which states that: in any sequence of adjacent and active neurons, the first and last neurons in the sequence should be maintained active. The second rule is the *Noise Elimination (NE)* , which calls for turning off an active neuron, in any given layer, surrounded by inactive neurons in the same layer; or an active neuron at the beginning/end of a layer followed/preceded by inactive neuron in the same layer. The POC and NE rules are accomplished through the lateral interaction and the 2/3 rule at the neuronal level. The idea of the 2/3 rule was introduced by Grossberg and Stone [7] and Grossberg and Carpenter [8] to show how an ART system can be primed by a previous event to expect a subsequent event that may or may not occur. In this work we employ a similar concept stated as follows: if the input signal from the image pixel and at least one lateral input signal from the neighboring neuron are active, then the neuron will fire or will remain active. If either condition is violated, then the neuron remains inactive or shuts off.

To illustrate the network operation philosophy, consider the state of the neurons shown in figure 6 just after stage 1. Note that by definition, the adjacent active neurons 3 through 5, represent a line segment. Also observe that the active neuron, surrounded by two inactive neurons, and the two active neurons at the beginning and end of the layer represent noise. As shown bellow during the second stage of operation, the network preserves the existing connectivity and eliminates any noise.

During the first cycle of stage 2 and after the calculation of the output of each neuron, the layer state will change from 101110101 to 001110000. The outputs of neurons 1 and 9 has changed due to the lateral inhibition signals sent from the neighboring inactive neurons (2 and 8). In addition, due to the lateral inhibition signals sent from neurons 6 and 8 to neuron 7, this neuron becomes inactive. Consequently, the noise is eliminated.

The lateral inhibition signals from neuron 2 to neuron 3, and from neuron 6 to neuron 5, will attempt to turn off neurons 3 and 5. But since each of these neurons receives two active input signals, therefore, by virtue of the 2/3 rule, neurons 3 and 5 will remain active. Thus, the inhibitory attempt to break down the existing connectivity is prevented. In principle, the 2/3 rule allows a connected sequence of active neurons to deter any external force to break down their connectivity to one another.

After the second cycle, the output of each neuron remains unchanged and, thus, it is said that the layer has reached its stable state. The above processes occur in parallel for each layer in a slab, and after two operation cycles each slab will detect a specific type of line segment.

3. Experimental Results

The proposed network were tested with various images containing one-pixel wide line segments. Example of these images are shown in figure 7a. In all images, the network successfully detected any sequence of adjacent black pixels oriented horizontally, vertically, or diagonally. The network also succeeded in detecting line segments surrounded by noise such as the image shown in figure 7b.

The networks error mainly occurred when the adjacency of a group of pixels yields a possibility of multiple paths. In such a case, the network detects a line segment that is not visually present in the image. This is illustrated by the patterns shown in figure 7c where each pattern is an L shape and, thus, it is composed of vertical and horizontal line segments. However, the two diagonal pixels also form a diagonal line and, consequently, will be detected by the diagonal slabs.

4. Discussion and Conclusion

In this paper we have presented our initial work for developing neural detectors. Each detector is a slab of neurons grouped into specific number of layers. The layers in a given slab are oriented in accordance with the slab type. As demonstrated above, the proposed method is original in that the whole image is taken as input. In addition, since there are no intra-slab or inter-slab connections, the information processing in each slab and in each layer in a slab may be performed in parallel. This feature reduces computation time and allows hardware implementation. Computation time can further be reduced by processing only the neurons with active signals from the image pixels.

When considering software implementation, the network operation can be modified as follows: given an input image matrix, create four copies of this matrix. Consider each copy as being a slab. Thereafter, calculate the output of each active neuron only once. Since inactive neurons can never change their state, by virtue of the 2/3 rule, therefore, they need not be processed.

In comparison with the sliding-window method, the proposed method is approximately 8 times faster. To illustrate this, consider an image containing 512x512 pixels. With the proposed method, there will be, at the maximum, 262144 additions and 262144 multiplication operations;

and with 3x3 sliding window there will be 2097152 additions and 2359296 multiplication. If each operation takes about 1µ second, then the time taken by the proposed method to process the whole image only once, is 0.524 second; and the time taken by using the 3x3 sliding window is 4.456 second.

Though the proposed method presented good results, the problem of multiple paths is yet to be solved In addition, the proposed method will be extended to detecting curves and circular shapes.

5. References

[1] Gonzalez, R. C. and Woods, R. E. Digital Image Processing. Addison Wesley, 1992.

[2] Bhatia, P. , Srinivasan, V. and Ong, S.H. Single Layer Edge Detector with Competitive Unsupervised Learning. Proc. IJCNN'91, Vol. 1, pp. 634-639.

[3] Etemad, K. and Chellapa, R. A Neural Based Edge Detector. Proc. IJCNN'93, Vol. 1, pp. 132-137.

[4] Moura, L. and Martin, F. Edge Detection Through Coperation and Competition. Proc. IJCNN'93, Vol. 3, pp. 2588-2593.

[5] Lepage, R. and Poussart, D. Multi-resolution Edge Detection Through Coperation and Competition. Proc. IJCNN'92, Vol. 4, pp. 438-443.

[6] von der Marlsberg, C. Self-organization and Orientation-sensitive Cells in the Striate Cortex. Kybernetik, Vol. 14, pp. 85-100. 1973.

[7] Grossberg, S. and Stone, G. Neural dynamics of word recognition and recall: Attentional priming, learning, and resonance. Psychological Review, 93, 46-74. 1986

[8] Carpenter, G. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, 37, 54-115. 1987

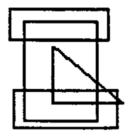

Figure 1. A hypothetical image.

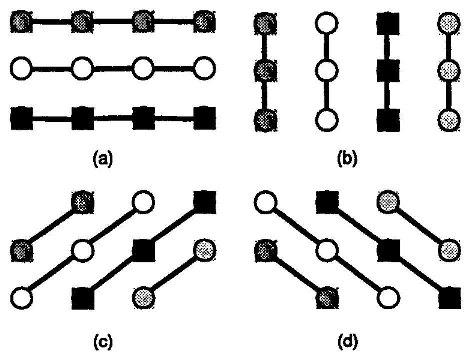

Figure 2. Slab structure. a) *h_slab*; b) *v_slab*; c) *d1_slab*; and d) *d2_slab*. Each circle indicates a neuron. Neurons in a layer are indicated by circles with the same texture. Solid lines indicate excitatory connections.

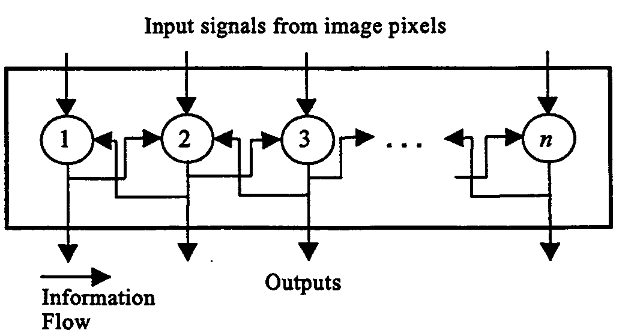

Figure 3. Connection between neurons in a given layer of any slap.

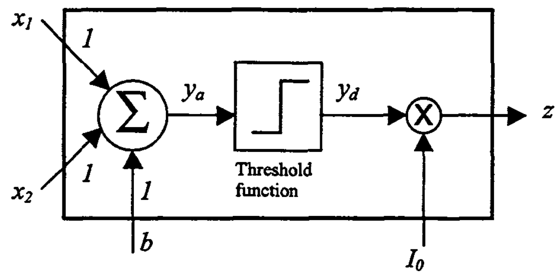

Figure 4. Mathematical model of the proposed neuron. The inputs x_1 and x_2 are lateral signals coming for two neighboring neurons; b is the bias input; I_0 is the input signal from an image pixel; and the internal signals y_a and y_d are, respectively, analog and discrete signals. The symbol ⊗ indicates a multiplication operation.

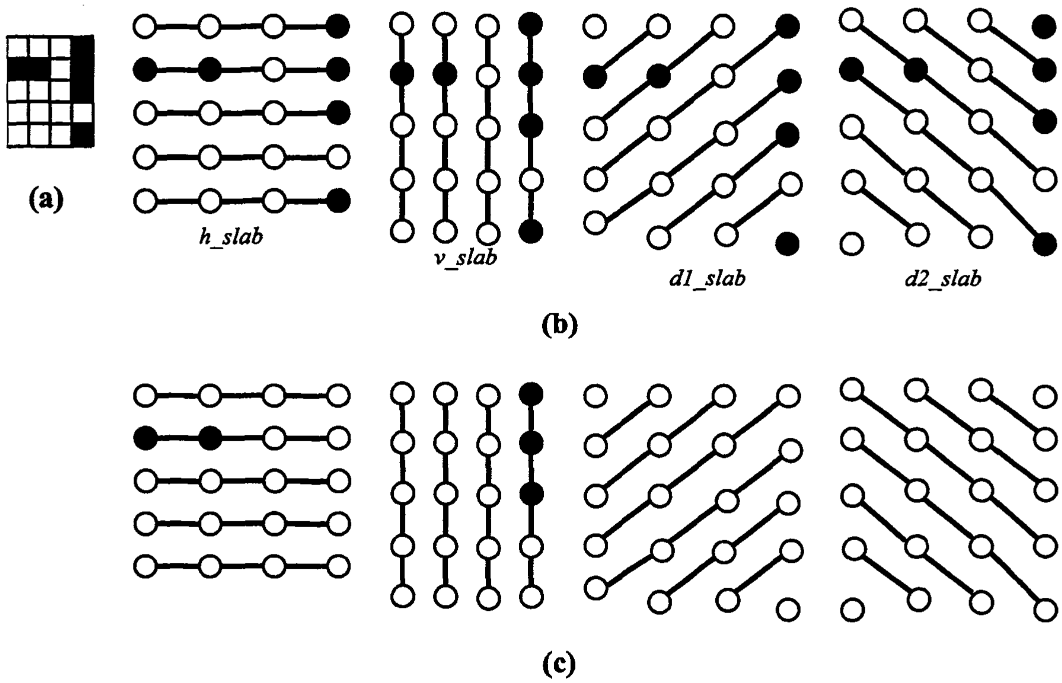

Figure 5. Operation of the Proposed Network. a) Input image. b) The state of each slab after the first stage of operation. At this stage each slab becomes a replica of the input image. c) The state of each slab after the second stage of operation, after which each layer in a given type of slab detects a corresponding type of line segment, if such line exists in the input image.

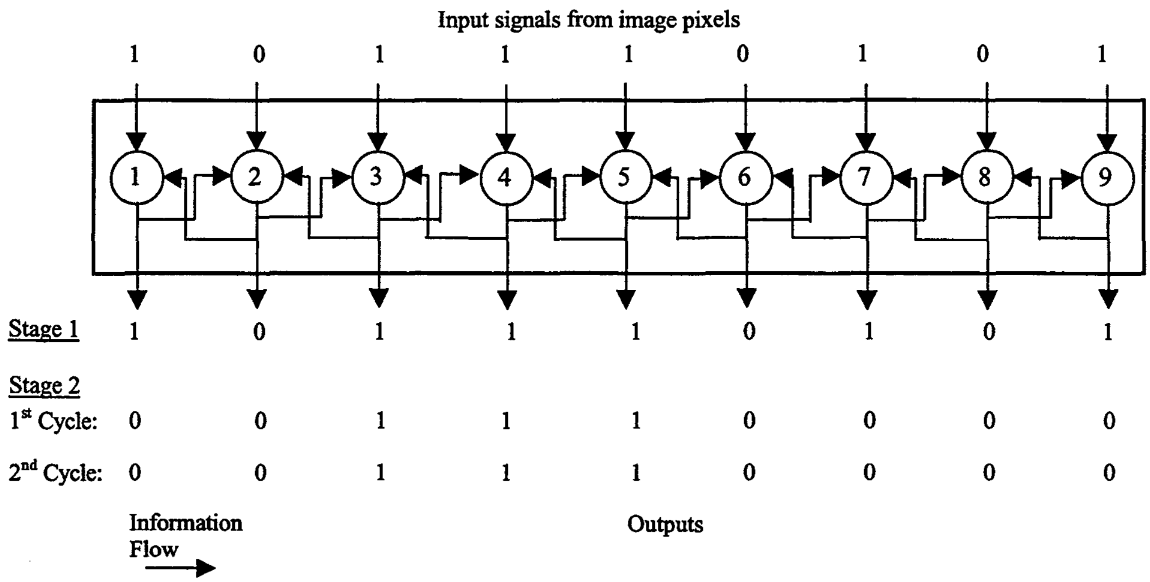

Figure 6. Illustration of the POC and NE rules (see section 2.3). After the first stage of operation, the output of the layer is a replica of the input signals. After the first cycle of the second stage, isolated active neurons (noise) become inactive, due to the lateral inhibition signals from neighboring inactive neurons. By virtue of the 2/3 rule, the connectivity of the three adjacent and active neurons is preserved. After the second cycle of the second stage, the layer output remains unchanged.

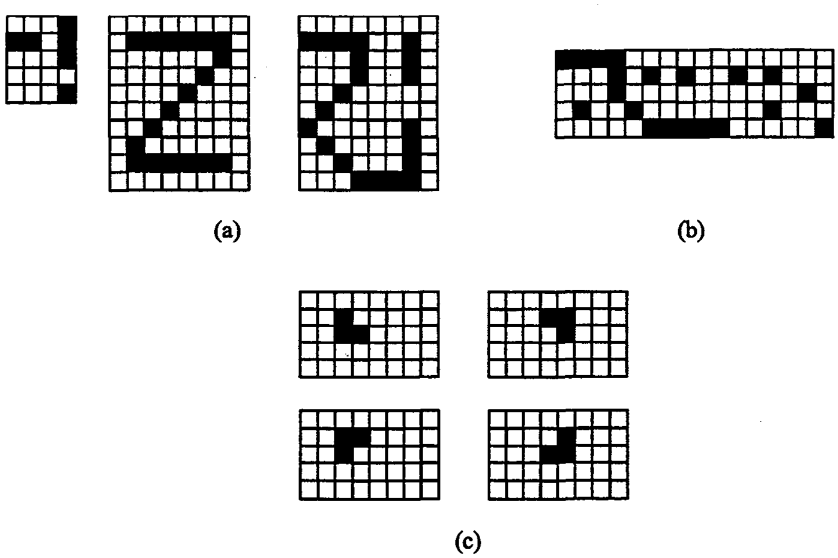

Figure 7. Examples of one-pixel wide test images. a) Image with line segments; b) Image with noise; c) pattern that yield multiple paths.